

**King Saud University**  
**College of Computer & Information Science**  
**CSC111 – Tutorial03**  
**IO, Variables, Expressions**  
**All Sections**

---

**Objectives:**

- 1- Student should learn how to define variable, assign them values and write arithmetic expressions.
- 2- Student should learn how to use class Scanner to read inputs.
- 3- Student should learn how to output results using `System.out.println`.
- 4- Student should learn how to read a problem statement and analyze it as following:
  - a. Find out if program needs input, how many inputs it is going to accept and of what type.
  - b. Decide if variables are needed, how many variable and of what type.
  - c. Understand the computation operations that are needed to solve the problem (i.e., if program needs to compute certain values using arithmetic expression).
  - d. Decide what is the program is going to output to the end user.
- 5- Student should learn how to write expressions and use operators according to precedence rules.

## Exercise 1

- 1) Show the result of the following code:
  - a) `System.out.println(2 * (5 / 2 + 5 / 2));`
  - b) `System.out.println(2 * 5 / 2 + 2 * 5 / 2);`
  - c) `System.out.println(2 * (5 / 2));`
  - d) `System.out.println(2 * 5 / 2);`
  
- 2) Are the following statements correct? If so, show the output.
  - a) `System.out.println("25 / 4 is " + 25 / 4);`
  - b) `System.out.println("25 / 4.0 is " + 25 / 4.0);`
  - c) `System.out.println("3 + 2 is " + 3 + 2);`
  - d) `System.out.println("3 + 2 is " + (3 + 2));`
  - e) `System.out.println("3 * 2 / 4 is " + 3 * 2 / 4);`
  - f) `System.out.println("3.0 * 2 / 4 is " + 3.0 * 2 / 4);`
  
- 3) Suppose **m** and **r** are integers. Write a Java expression for **mr<sup>2</sup>** to obtain a floating-point result.
  
- 4) How would you write the following arithmetic expression in Java?

a) 
$$\frac{4}{3(r + 34)} - 9(a + bc) + \frac{3 + d(2 + a)}{a + bd}$$

b) 
$$5.5 \times (r + 2.5)^{2.5 + t}$$

- 5) Show the output of the following code:

```
double a = 6.5;
a += a + 1;
System.out.println(a);
a = 6;
a /= 2;
System.out.println(a);
```

- 6) Show the output of the following code:

```
int a = 6;
int b = a++;
System.out.println(a);
```

```
System.out.println(b);
a = 6;
b = ++a;
System.out.println(a);
System.out.println(b);
```

7) Is the following code going to print **0.1** and why?

```
System.out.println(1.0 - 0.9);
```

## Solution

**Note: Unanswered questions are left as self-exercises for student.**

1)

- a) 8
- b) 10

2)

- a) 25 / 4 is 6
- b) 25 / 4.0 is 6.25
- c) 3 + 2 is 32
- d) 3 + 2 is 5

3) **double** x = m \* Math.pow(r,2);

4)

```
a) double result = 4 / (3 * (r + 34)) - 9 * ( a + b * c)
+ (3 + d * ( 2 + a)) / (a + b * d);
```

5)

14.0
3.0

6)

7
6
7
7

7) The output is going to be **0.09999999999999998** not **0.1** cause double type is not stored with complete accuracy but instead approximated. This is called *rounding error* and it is why double can not be used for applications that need decimal point with precision such as financial applications.

## Exercise 2

Write a program that changes a given amount of money into smaller monetary units. The program lets the user enter an amount as a **double** value representing a total in Riyals and Halals, and outputs a report listing the monetary equivalent in the maximum number of riyals, halves (SR 0.5), quarters (SR 0.25), qirsh (SR 0.5), and halalah (SR 0.01), in this order, to result in the minimum number of coins.

Here is a sample run:

```
Enter an amount, for example, 11.88: 11.88 ↵
Your amount 11.56 consists of
    11 riyals
     1 halves
     1 quarters
     2 qirshs
     3 halalahs
```

## Solution

Here are the steps in developing the program:

- A. Prompt the user to enter the amount as a decimal number, such as **11.88**.
- B. Convert the amount (e.g., **11.88**) into halalas (**1188**).
- C. Divide the halalas by **100** to find the number of riyals. Obtain the remaining halalas using the halalas remainder **100**.

- D. Divide the remaining halalas by **50** to find the number of halves.  
Obtain the remaining halalas using the remaining halalas remainder **50**.
- E. Divide the remaining halalas by **25** to find the number of quarters.  
Obtain the remaining halalas using the remaining halalas remainder **25**.
- F. Divide the remaining halalas by **5** to find the number of qirshs.  
Obtain the remaining halalas using the remaining halalas remainder **5**.
- G. The remaining halalas are the halalas (can not be divided).
- H. Display the result.

```
import java.util.Scanner;

public class ComputeChange {
    public static void main(String[] args) {
        // Create a Scanner
        Scanner input = new Scanner(System.in);

        // Receive the amount
        System.out.print("Enter an amount in double, for
example 11.88: ");
        double amount = input.nextDouble();
        int remainingAmount = (int) (amount * 100);

        // Find the number of one riyals
        int numberOfOneRiyals = remainingAmount / 100;
        remainingAmount = remainingAmount % 100;

        // Find the number of halves in the remaining amount
        int numberOfHalves = remainingAmount / 50;
        remainingAmount = remainingAmount % 50;

        // Find the number of quarters in the remaining amount
        int numberOfQuarters = remainingAmount / 25;
```

```

remainingAmount = remainingAmount % 25;

// Find the number of qirshs in the remaining amount
int numberOfQirshs = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of halalas in the remaining amount
int numberOfHalalas = remainingAmount;

// Display results
System.out.println("Your amount " + amount + " consists
of");
System.out.println("    " + numberOfOneRiyals + "
riyals");
System.out.println("    " + numberOfHalves + " halves ");
System.out.println("    " + numberOfQuarters + "
quarters");
System.out.println("    " + numberOfQirshs + "
qirshs");
System.out.println("    " + numberOfHalalas + "
halalas");
}
}

```

**Done...**